

**REMARKS**

Claims 1-50 are pending, of which Claims 1, 26, 49 and 50 are independent. All claims have been rejected under 35 U.S.C. §102. For the reasons described below, all claims are in condition for allowance.

**Rejections Under 35 U.S.C. §102(e)**

Claims 1-50 were rejected under §102(e) based on U.S. Patent No. 7,043,636 to Smeets (hereinafter, "Smeets").

Disclosed embodiments of the present invention relate to a method for identifying protected software while it is being executed using a superfingerprint. To create the superfingerprint, while the protected software is executing, a supervising program selects specific portions of the protected software and results of executing said protected software. The supervising program performs computations on the selected portions of the executing protected software and of the results of executing the protected software to obtain a collection of fingerprints. The collection of fingerprints is combined by the supervising program to create a superfingerprint of the protected software.

These superfingerprints are then used to identify the protected software whenever it is being executed. Namely, the superfingerprint for a protected software is stored, say on a user device. When a software application is executing, a supervising program on the device selects portions of this executing software, performs computations on those portions to produce fingerprints and compares the collection of these computed fingerprints with the stored superfingerprint for the protected software. The supervising program declares the protected software to be the same as the executing software if an approximate match is found.

Smeets relates to a scheme to maintain the integrity of stored code, ensuring that the code text intended to be present in a device is present, and protect certain dynamic data values from change at will by an unscrupulous user. Smeets teaches a method of generating hash signatures

for certain data values. To protect static data, Smeets uses an audit hash value calculated by using a hash algorithm. This hash algorithm may be a function of a static parameter, such as a serial number. To protect dynamic data, Smeets uses a first dynamic audit hash value, calculated by a first dynamic hash algorithm, and, to protect the dynamic authorization data, a second dynamic audit hash value, calculated by a second dynamic has algorithm.

Smeets teaches performing a hash function on protected code text when the latter is not being executed, so Smeets' hash values are derived from the unexecuted code text, not from executing the protected software and computing fingerprints from the protected software. What Smeet does execute is a hash function, but the hash function is not protected itself. . Moreover, the hash values created by Smeets are a function of the code text and are not from executing the protected software. Further, the audit hash values created by Smeets are not superfingerprints created by a supervising program selecting specified portions of the executing protected software.

Disclosed embodiments of Applicants' invention identify a protected software or software group by looking at selected portions of at least one of said executing protected software and of results of said executing protected software. By contrast, Smeets looks at the entire body of code text, in regard to static data, and looks at the user-visible "audit hash" values, in regard to dynamic data, not the executing protected software and results of the executing software which need not be visible.

By focusing on the software while it is executing, the invention is able to address the situation where an unscrupulous user tries to camouflage unauthorized protected software (program P) into a larger software program (program C). For instance, the unscrupulous user takes program P and embeds it into program C. A system of the Smeet type would compare the code text of P and C and would not find any similarities because Smeet looks at the code text. Smeets is concerned with maintaining the integrity of code text, and therefore, Smeet wants to ensure that the text that is supposed to be present is in fact present. Smeets also wants to be sure

that certain dynamic data values are not changed at will by an unscrupulous user. For example, Smeets' system ensures that cell phone data has not been tampered with.

Conversely, the supervising program of the present invention would be able to detect the similarities between P and C because the claimed invention analyzes the software while it is executing. By analyzing the protected software while it is executing, the present invention is able to address such situations where an unscrupulous user tries to camouflage unauthorized code by embedding it into a larger program.

As such, Smeets does not address the problems associated with unscrupulous users embedding unauthorized software into larger programs and, therefore, Smeets does not discuss the solutions to this problem set forth in claimed invention.

Thus, with regard to the Examiner's rejection of Claims 1, Smeets does not teach a method for creating a superfingerprint for identifying a software. Smeets makes no mention of executing said protected software at least once or, in each execution, using a supervising program, selecting specified portions of at least one of said executing software and of results of executing said protected software. Further, Smeets does not mention, in each execution, using a supervising program, performing computations on said selected portions to obtain a collection of fingerprints or combining, using the supervising program, said collections of fingerprints found in each execution into the superfingerprint of said protected software according to a combining rule. Therefore, the Examiner's rejection of Claim 1 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 2, Smeets makes no mention of executing the protected software a plurality of times. Further, Claim 2 is dependent on Claim 1 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 2 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 6, Smeets makes no mention of several protected softwares, let alone a fingerprint belonging to the superfingerprints of several protected softwares. Further, Claim 6 is dependent on Claim 1 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 6 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 16, Smeets makes no mention of selected portions, let alone a computation that involves only parts of said selected portions. Further, Claim 16 is dependent on Claim 1 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 16 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 20, Smeets makes no mention of selected portions or the execution of protected software. Further, Claim 20 is dependent on Claim 1 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 20 is overcome and reconsideration is respectfully requested.

Claims 3-5, 7-15, 17-19 and 21-25 are dependent on Claim 1 and therefore contain all the limitations of the base claim. Therefore, the Examiner's rejection is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 26, Smeets does not teach a method for identifying a first protected software. Smeets makes no mention of storing previously created superfingerprints for at least one protected software, executing said first protected software at least once or selecting by a supervising program selected portions of at least one of said executing protected software and of the results of executing said protected first software on each execution. Further, Smeets does not mention performing by said supervising program specified computations on said selected portions to obtain a collection of fingerprints, comparing said collection of fingerprints to said previously computed superfingerprint of at least one second protected software to determine whether there is an approximate match or declaring said the first

software to be the same as said second protected software if an approximate match is found. For these reasons, the Examiner's rejection of Claim 26 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 27, Smeets makes no mention of specified portions of executing protected software. Further, Claim 27 is dependent on Claim 26 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 27 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 33, Smeets makes no mention of determining whether the amount of commonality between said fingerprints of said first protected software and the fingerprints comprising said superfingerprint of said at least one second protected software exceeds a specified threshold in which case the first protected software is identified to be the same as the second protected software. Further, Claim 33 is dependent on Claim 26 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 33 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 38, Smeets makes no mention of selected portions, let alone a computation that involves only parts of said selected portions. Further, Claim 38 is dependent on Claim 26 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 38 is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 42, Smeets makes no mention of selected portions or the execution of protected software. Further, Claim 42 is dependent on Claim 26 and therefore contains all the limitations of the base claim. For these reasons, the Examiner's rejection of Claim 42 is overcome and reconsideration is respectfully requested.

Claims and 28-32, 34-37, 39-41 and 43-48 are dependent on Claim 26 and therefore contain all the limitations of the base claim. Therefore, the Examiner's rejection is overcome and reconsideration is respectfully requested.

With regard to the Examiner's rejection of Claim 49, Smeets does not teach a method for identifying a protected software group of a first software. Smeets makes no mention of storing previously created superfingerprints for at least one protected software group, executing said first protected software at least once or selecting portions of said executing first protected software and the results of executing said first protected software on each execution. Further, Smeets does not perform specified computations on said selected portions to obtain a collection of fingerprints, does not compare said collection of fingerprints to said previously computed superfingerprint of at least one second protected software group to determine whether there is an approximate match or declare said first protected software to be a member of said second protected software group if an approximate match is found. For these reasons, the Examiner's rejection of Claim 49 is overcome and reconsideration is respectfully requested.

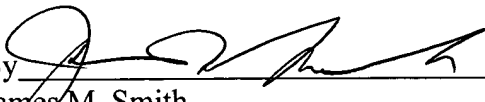
With regard to the Examiner's rejection of Claim 50, Smeets does not teach a method for identifying a protected software that is a member of a group of protected software. Smeets makes no mention of storing previously created superfingerprints for at least one protected software group and for members of that group, executing said protected software at least once, or selecting specified portions of said executing protected software and of the results of executing said protected software on each execution. Further, Smeets makes no mention of performing computation of said selected portions to obtain a collection of fingerprints, comparing said collection of fingerprints to said previously computed superfingerprint of at least one second protected software group and the superfingerprints of the members of said second protected software group to determine whether there is an approximate match with said group and at least one of said superfingerprints of said members of said group or declare said protected software to be the same as a particular member of said second protected software group if an approximate match is found. For these reasons, the Examiner's rejection of Claim 50 is overcome and reconsideration is respectfully requested.

**CONCLUSION**

In view of the above amendments and remarks, it is believed that all claims are in condition for allowance, and it is respectfully requested that the application be passed to issue. If the Examiner feels that a telephone conference would expedite prosecution of this case, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

HAMILTON, BROOK, SMITH & REYNOLDS, P.C.

By   
James M. Smith  
Registration No. 28,043  
Telephone: (978) 341-0036  
Facsimile: (978) 341-0136

Concord, MA 01742-9133

Dated: 12/4/6